Remap of solenoidal fields on unstructured quadrilateral grids

Five-Laboratory Conference on Computational Mathematics, Vienna, Austria, 19-23 June 2005

Pavel Bochev

Computational Mathematics and Algorithms

Sandia National Laboratories

Mikhail Shashkov

Mathematical Modeling and Analysis T-7

Los Alamos National Laboratory

Supported in part by

MICS

Mathematical, Information, and Computational Sciences

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04–94AL85000.

What is remap and where is it needed?

Remap = transfer of data between grids, subject to constraints

Remap is ubiquitous in computational modeling:

– ALE methods	remap after rezoning
– Coupled multiphysics	FE Navier-Stokes + FV transport
– Multiscale methods	atomistic-to-continuum coupling
– Multilevel methods	restriction and prolongation

Our focus: remap of div-free fields on unstructured grids

- ❑ **Coupled physics:** Remap = constrained interpolation
  - – Lagrange multipliers → **global** & **implicit**
- ❑ **ALE methods:** Remap = advection
  - – Transport algorithms → **local** & **explicit**

Remap between structured grids:

relies upon grid structure and is not of interest to us

Nomenclature

Definitions and notations:

Bochev and Hyman, Application of algebraic topology to compatible spatial discretizations, *Proceedings of the Five-Laboratory Conference on Computational mathematics*, Vienna, June 19-23, 2005.

Computational grid: **chain complex**

$$\partial\partial = 0 \quad C_0 \xleftarrow{\partial} C_1 \xleftarrow{\partial} C_2 \xleftarrow{\partial} C_3$$

$$\langle c^k, c_k \rangle$$ **duality**

Field representation: **cochain complex**

$$\delta\delta = 0 \quad C^0 \xrightarrow{\delta} C^1 \xrightarrow{\delta} C^2 \xrightarrow{\delta} C^3$$

We encode divergence free fields as **2-cochains**:

$$\mathbf{B}: \quad \nabla \cdot \mathbf{B} = 0 \quad \rightarrow \quad b \in C^2; \quad \delta b = 0$$

Sandia National Laboratories

Formal statement of the remap problem

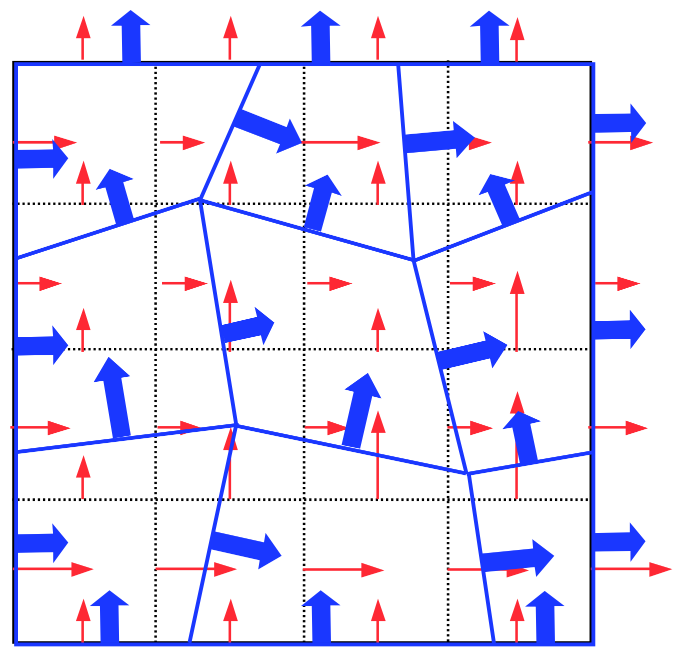
Given $\mathcal{K} = (C_0, C_1, C_2, C_3)$ “old” cell complex with cochains $C = (C^0, C^1, C^2, C^3)$

$\tilde{\mathcal{K}} = (\tilde{C}_0, \tilde{C}_1, \tilde{C}_2, \tilde{C}_3)$ “new” cell complex with cochains $\tilde{C} = (\tilde{C}^0, \tilde{C}^1, \tilde{C}^2, \tilde{C}^3)$

$b \in C^2; \quad \delta b = 0$ Solenoidal cochain on old complex (2-cocycle)

Find $\tilde{b} \in \tilde{C}^2 : \begin{cases} \delta \tilde{b} = 0 \\ \tilde{b} \approx b \end{cases}$ Solenoidal cochain on the new complex that approximates *b*

We focus on two-dimensions and quad cells:



A good remapper

- Is accurate
- Preserves energy
- Has good feature retention
- Does not “ring” at jumps
- Is efficient (EXPLICIT)!

Lagrange Multipliers Solution

Viewpoint: Remap = constrained interpolation

Constrained Optimization problem

$$J(\tilde{b};b) = \frac{1}{2}\left\|b-\tilde{b}\right\|^2 \quad \longrightarrow \quad \min_{\tilde{b}\in\tilde{C}^2} J(\tilde{b};b) \quad \text{subject to} \quad \delta\tilde{b}=0$$

Lagrangian functional

$$\min_{\tilde{b}\in\tilde{C}^2}\max_{\tilde{\lambda}\in\tilde{C}^3} J(\tilde{b};b) - (\delta\tilde{b},\tilde{\lambda})_{\tilde{C}^3} \quad \longrightarrow \quad \begin{aligned} (\tilde{b},\tilde{a})_{\tilde{C}^2} - (\delta\tilde{a},\tilde{\lambda})_{\tilde{C}^3} &= (b,\tilde{a}) \quad && \forall \tilde{a}\in\tilde{C}^2 \\ (\delta\tilde{b},\tilde{\mu})_{\tilde{C}^3} &= 0 && \forall \tilde{\mu}\in\tilde{C}^3 \end{aligned}$$

Advantages

- **Arbitrary new and old grids**

Disadvantages

- **Implicit and global:** requires inversion of a saddle-point matrix

$$\begin{pmatrix} \mathbf{M} & \mathbf{B} \\ \mathbf{B}^T & 0 \end{pmatrix}\begin{pmatrix} \tilde{b} \\ \tilde{\lambda} \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$$

*Carey et al. IJNME 50, **2001**, Girault, Scott, Calcolo **2003***

Transport Solution

Viewpoint: Remap = constrained transport

Advection equation

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \left(\mathbf{B} \times \mathbf{u}_{REL}\right)$$

$$\mathbf{E} = \mathbf{B} \times \mathbf{u}_{REL}$$

Invariant form

$$\partial_t b = -d * \left(*b \wedge \mathbf{u}_{REL}\right)$$

$$e = *\left(*b \wedge \mathbf{u}_{REL}\right)$$

Cochain translation

$$\delta_t b = -\delta *_h \left(*_h b \wedge \mathbf{u}_{REL}\right)$$

$$e = *_h\left(*_h b \wedge \mathbf{u}_{REL}\right)$$

CT Update of *b*

$$\tilde{b} = b - \Delta t\, \delta e \quad \Rightarrow \quad \delta\tilde{b} = \delta b - \Delta t\, \delta\delta e = \delta b \quad \Rightarrow \quad \delta b = 0 \;\Rightarrow\; \delta\tilde{b} = 0$$

If the old field was solenoidal,the new field stays solenoidal

Disadvantages

- old and new grid must have the same topology
- discrete $*$ operation difficult for unstructured grids

Advantages

- Explicit and local

Based on CT scheme of Evans, Hawley, The Astrophysical Journal 332, 1988

Sandia National Laboratories

Reconstruction to virtual edges

$$e = *_h\left(*_h b \wedge \mathbf{u}_{REL}\right)$$

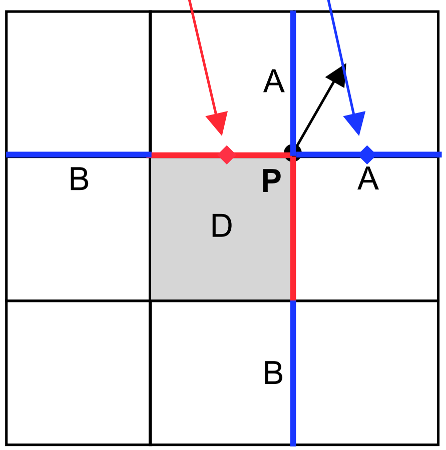
$$*_h b \rightarrow \mathbf{B}(\mathrm{P})$$

$$\left(*_h b \wedge \mathbf{u}_{REL}\right) \rightarrow \left(\mathbf{u}_2\mathbf{B}_1 - \mathbf{u}_1\mathbf{B}_2\right)(\mathrm{P})$$

$$*_h\left(*_h b \wedge \mathbf{u}_{REL}\right) \rightarrow \left(\mathbf{u}_2\mathbf{B}_1 - \mathbf{u}_1\mathbf{B}_2\right)(\mathrm{P})$$

Reconstruction of B = averaging to P

$$\mathbf{B}_i(\mathrm{P}) = \left(b_{\text{Donor}} + b_{\text{Ahead}}\right)/2$$



Advection requires upwind interpolation at P:

$$b_{\text{Ahead}} = b_{\text{Donor}} + S\left(\frac{\left|b_{\text{Donor}}\right| + \left|b_{\text{Ahead}}\right|}{2} - \Delta t \mathbf{u}_i\right)$$

S ∈ {mono,harmonic, Van Leer, Donor}

Extension to unstructured grids:

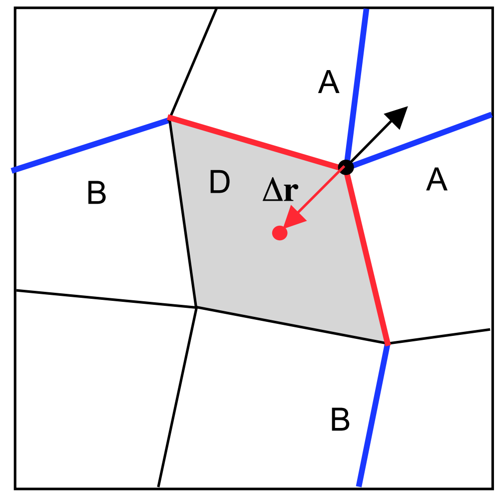
A. Robinson, P. Bochev, P. Rambo.
`http://infoserve.sandia.gov/sand_doc/2001/012146p.pdf`

CT Remap simply builds a Taylor expansion

$$\mathbf{B}_1(\mathbf{r}+\Delta\mathbf{r}) = \mathbf{B}_1(\mathbf{r}) + \frac{\partial \mathbf{B}_1(\mathbf{r})}{\partial x}\Delta x + \frac{\partial \mathbf{B}_1(\mathbf{r})}{\partial y}\Delta y + O\left(\left|\Delta\mathbf{r}\right|^2\right)$$

$$\mathbf{B}_2(\mathbf{r}+\Delta\mathbf{r}) = \mathbf{B}_2(\mathbf{r}) + \frac{\partial \mathbf{B}_2(\mathbf{r})}{\partial x}\Delta x + \frac{\partial \mathbf{B}_2(\mathbf{r})}{\partial y}\Delta y + O\left(\left|\Delta\mathbf{r}\right|^2\right)$$

$$\boxed{\nabla \cdot \mathbf{B} = \frac{\partial \mathbf{B}_1}{\partial x} + \frac{\partial \mathbf{B}_2}{\partial y} = 0}$$



$$\mathbf{B}_1(\mathbf{r}+\Delta\mathbf{r}) = \mathbf{B}_1(\mathbf{r}) + \left(\cancel{\frac{\partial \mathbf{B}_1(\mathbf{r})}{\partial x}\Delta x + \frac{\partial \mathbf{B}_2(\mathbf{r})}{\partial y}\Delta x}\right) + \left(\frac{\partial \mathbf{B}_1(\mathbf{r})}{\partial y}\Delta y - \frac{\partial \mathbf{B}_2(\mathbf{r})}{\partial y}\Delta x\right) + O\left(\left|\Delta\mathbf{r}\right|^2\right)$$

$$\mathbf{B}_2(\mathbf{r}+\Delta\mathbf{r}) = \mathbf{B}_2(\mathbf{r}) + \left(\frac{\partial \mathbf{B}_2(\mathbf{r})}{\partial x}\Delta x - \frac{\partial \mathbf{B}_1(\mathbf{r})}{\partial x}\Delta y\right) + \left(\cancel{\frac{\partial \mathbf{B}_2(\mathbf{r})}{\partial y}\Delta y + \frac{\partial \mathbf{B}_1(\mathbf{r})}{\partial x}\Delta y}\right) + O\left(\left|\Delta\mathbf{r}\right|^2\right)$$

$$\mathbf{B}(\mathbf{r}+\Delta\mathbf{r}) = \mathbf{B}(\mathbf{r}) + \begin{pmatrix} \dfrac{\partial}{\partial y}\left(\mathbf{B}_1(\mathbf{r})\Delta y - \mathbf{B}_2(\mathbf{r})\Delta x\right) \\ -\dfrac{\partial}{\partial x}\left(\mathbf{B}_1(\mathbf{r})\Delta y - \mathbf{B}_2(\mathbf{r})\Delta x\right) \end{pmatrix} + O\left(\left|\Delta\mathbf{r}\right|^2\right) = \mathbf{B}(\mathbf{r}) - \nabla \times \left(\Delta\mathbf{r} \times \mathbf{B}\right)$$

Reconstruction must be exact for the 1st derivatives to get 2nd order accuracy.

$$\boxed{\mathbf{B}(\mathbf{r}+\Delta\mathbf{r}) = \mathbf{B}(\mathbf{r}) - \nabla \times \left(\Delta\mathbf{r} \times \mathbf{B}\right) + O\left(\left|\Delta\mathbf{r}\right|^2\right)}$$

Sandia National Laboratories

Constrained Interpolation Algorithm

Principal idea: exploit **existence** of discrete potentials in exact sequences
$\Rightarrow$ **divergence-free constraint automatically satisfied**

Key component: an **explicit** potential recovery algorithm

RECOVERY **REMAP**

$$\begin{tikzcd}
\delta b = 0 \arrow[d] & & \delta \tilde{b} = 0 \\
b = \delta a \arrow[d] & & \tilde{b} = \delta \tilde{a} \arrow[u] \\
a \arrow[rr] & & \tilde{a} \arrow[u]
\end{tikzcd}$$

POSTPROCESSING

$\mathcal{R}(a) \mapsto$

Patch recovery
PP operators

"reconstruction"

OPTIMIZATION

$$a(\lambda) = \lambda a + (1-\lambda)\mathcal{R}(a)$$

$$\lambda_{opt} = \arg\min \sum_{\tilde{K}\in\tilde{C}_2} \left(\|\delta a\|^2 - \|\delta \mathcal{I}a(\lambda)\|^2 \right)^2$$

"limiting"

INTERPOLATION

$$\tilde{a} = \mathcal{I}a(\lambda_{opt})$$

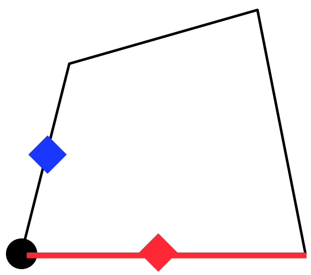
"transport"

Sandia National Laboratories

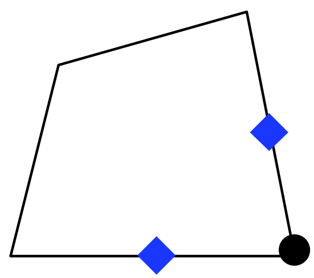
Computation of coboundary

$$a \in C^1 \quad \Rightarrow \quad a = \sum_{i,j=0,1} a_{ij}\sigma^1_{ij}$$

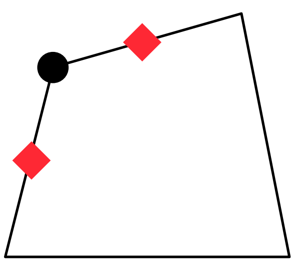
$$b \in C^2 \quad \Rightarrow \quad b = \sum_{F=D,U,R,L} b_F \sigma^2_F$$



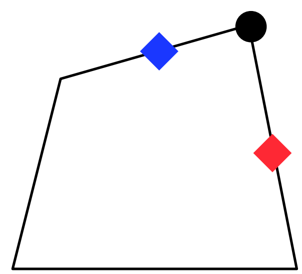
$$\delta\sigma^1_{00} = \sigma^2_D - \sigma^2_L$$



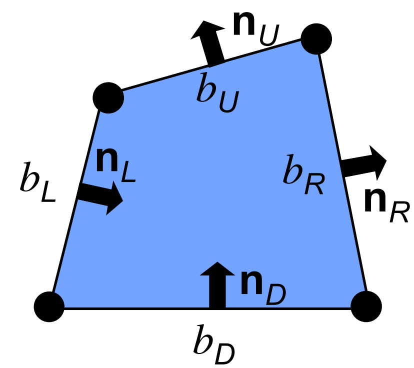
$$\delta\sigma^1_{10} = -\sigma^2_R - \sigma^2_D$$



$$\delta\sigma^1_{01} = \sigma^2_L + \sigma^2_U$$



$$\delta\sigma^1_{11} = \sigma^2_R - \sigma^2_U$$



$$\delta a = a_{00}\left(\sigma^2_D - \sigma^2_L\right) + a_{10}\left(-\sigma^2_R - \sigma^2_D\right) + a_{01}\left(\sigma^2_L + \sigma^2_U\right) + a_{11}\left(\sigma^2_R - \sigma^2_U\right)$$

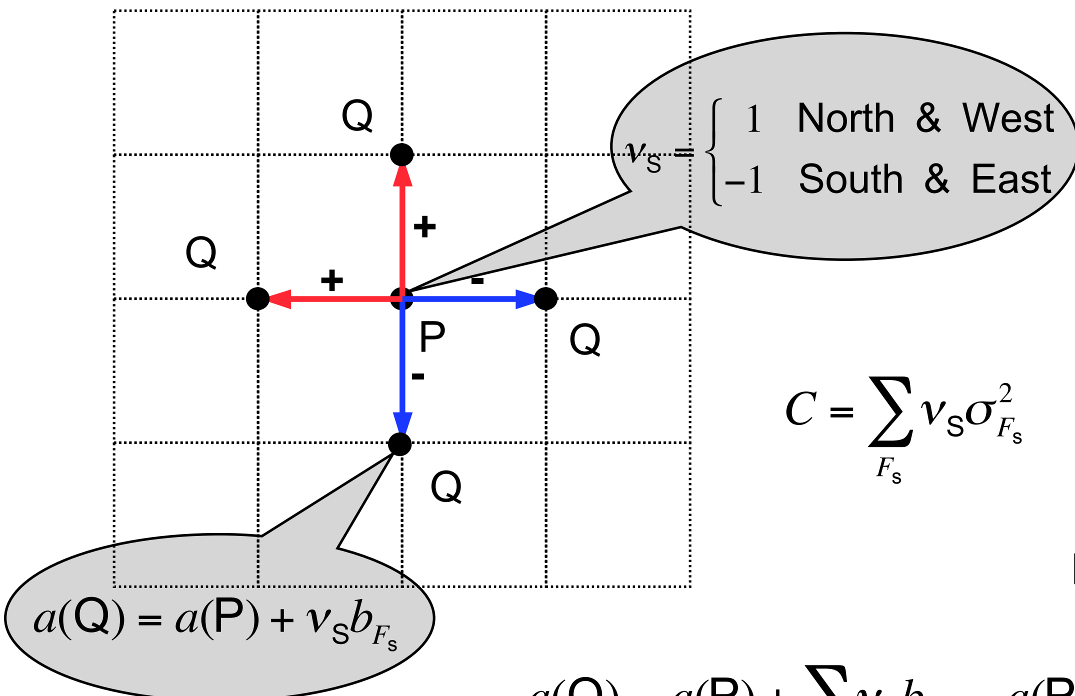
$$\Downarrow$$

$$\delta a = \left(a_{00} - a_{10}\right)\sigma^2_D + \left(a_{01} - a_{11}\right)\sigma^2_U + \left(a_{11} - a_{10}\right)\sigma^2_R + \left(a_{01} - a_{00}\right)\sigma^2_L$$

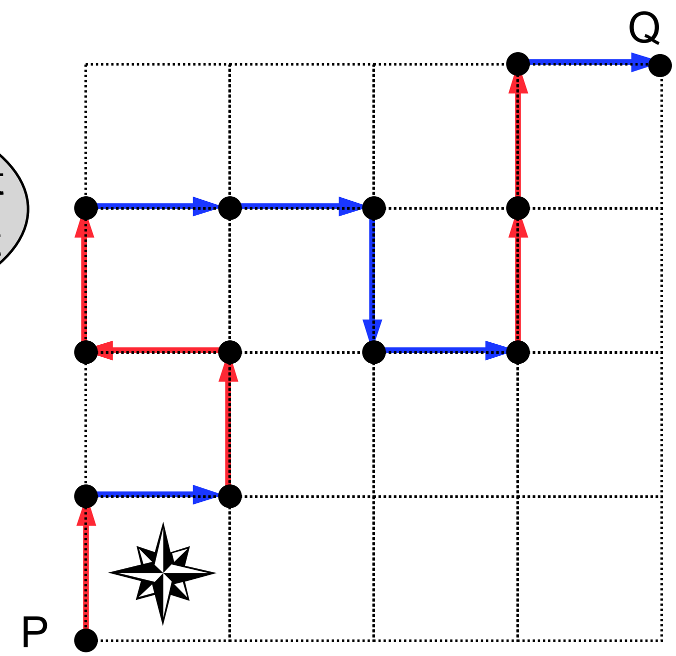
Sandia National Laboratories

Explicit Potential Recovery

$$\left.\begin{aligned}\delta a &= (a_{00}-a_{10})\sigma_D^2 + (a_{01}-a_{11})\sigma_U^2\\ &\quad +(a_{11}-a_{10})\sigma_R^2 + (a_{01}-a_{00})\sigma_L^2\\ b &= b_D\sigma_D^2 + b_U\sigma_U^2 + b_R\sigma_R^2 + b_L\sigma_L^2\end{aligned}\right\} \Rightarrow \delta a = b \Leftrightarrow \begin{cases} b_D = (a_{00}-a_{10}) & b_L = (a_{01}-a_{00})\\ b_U = (a_{01}-a_{11}) & b_R = (a_{11}-a_{10})\end{cases}$$



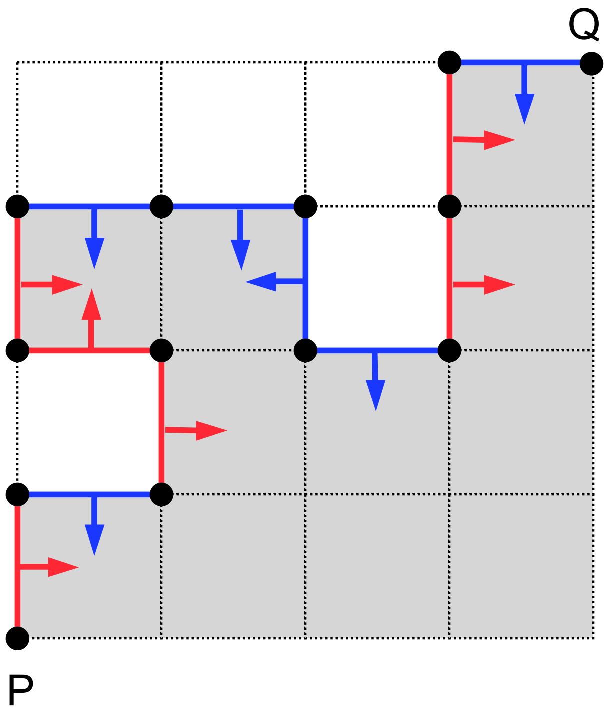
$$C = \sum_{F_s} \nu_S \sigma_{F_s}^2$$



$$a(\mathrm{Q}) = a(\mathrm{P}) + \sum_{F_s} \nu_S b_{F_s} = a(\mathrm{P}) + \langle b, C\rangle$$

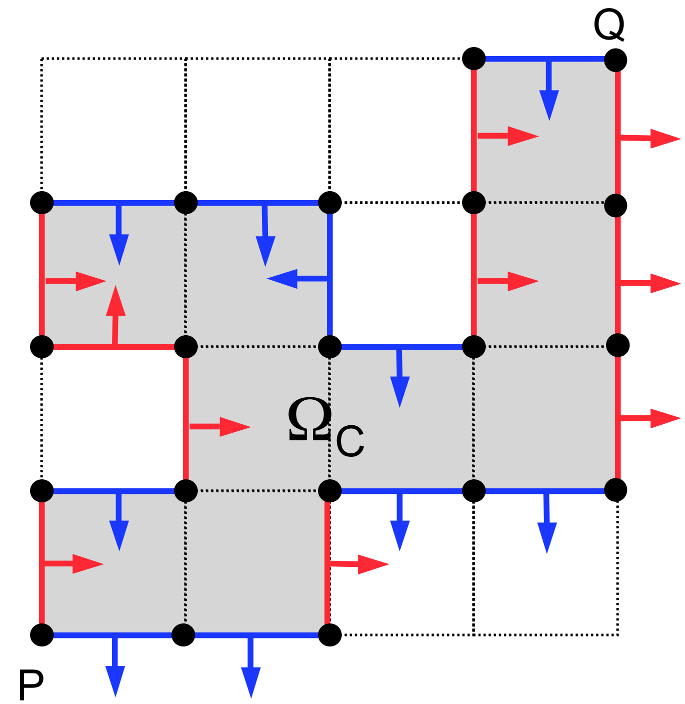
Sandia National Laboratories

Path Independence



$$C_1 = \sum_{S_1} \nu_{S_1} \sigma_{S_1}^2$$

$$C_2 = \sum_{S_2} \nu_{S_2} \sigma_{S_2}^2$$



$$\left.\begin{array}{r} \partial\Omega_C = C_2 - C_1 \\ \delta b = 0 \end{array}\right\} \Rightarrow 0 = \langle \delta b, \Omega_C \rangle = \langle b, \partial\Omega_C \rangle$$

$$\Rightarrow 0 = \langle b, C_2 - C_1 \rangle \Rightarrow \boxed{\langle b, C_2 \rangle = \langle b, C_1 \rangle = a(Q)}$$

Sandia National Laboratories

Recovery on Logically Rectangular Grids

1. Choose a spanning tree

$$C_{sp} = \sum_{S_{sp}} \nu_{S_{sp}} \sigma^2_{S_{sp}}$$

2. Initialize the root (gauge)

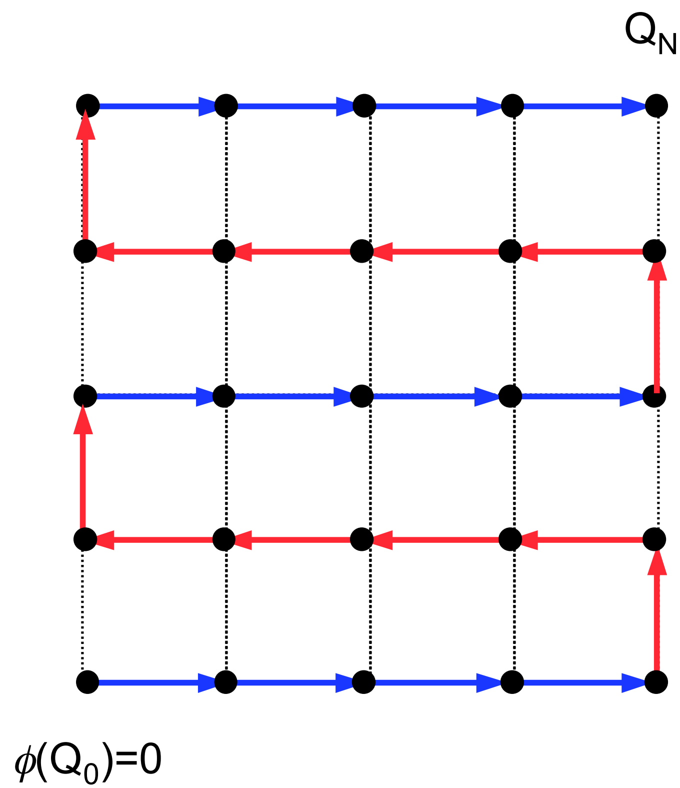
$$a(Q_0) = 0$$

3. Traverse the branches

```
for k=1:N
```

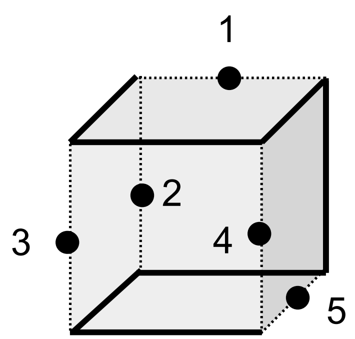
$$a(Q_k) = a(Q_{k-1}) + \nu_k b_{F_k}$$

```
end
```



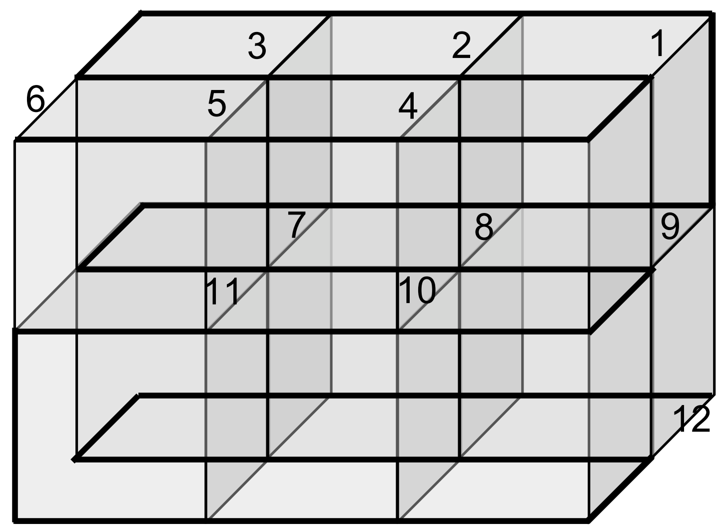
Base potential recovery in 3D

On one cell:



6 flux DOF - 1 constraint = 5
12 edges - 7SP edges = 5

On a logically rectangular mesh



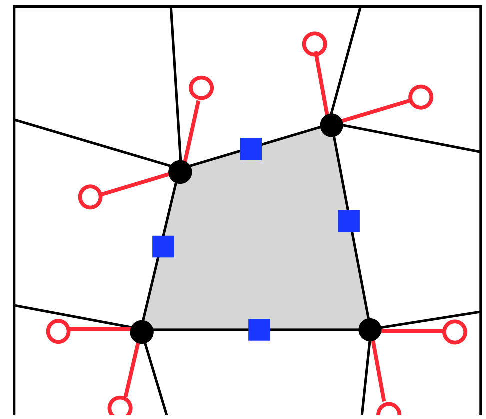
1. Choose a spanning tree and mark edges on co-spanning tree as **free**
2. Order faces with respect to the number of free edges
3. Recover $\mathbf{A}_B$ on all faces with 1 free edge; update number of free edges
4. If no faces with free edges left **then** stop, **else** proceed to step 3

Sandia National Laboratories

Post-processing and Optimization

Post-processing

- A_B → **base** Q1 (bilinear interpolant)
- A_E → **extended** 8 node serendipity



Optimization

$$\mathbf{A}\ (\lambda) = \lambda \mathbf{A} + (1-\lambda)\mathcal{R}(\mathbf{A})$$

$$\lambda_{opt} = \arg\min J(\mathbf{A}, \mathbf{A}(\lambda)$$

Requires conservative remap of magnetic energy (see *Shashkov and Margolin, LANL LAUR-2002*)

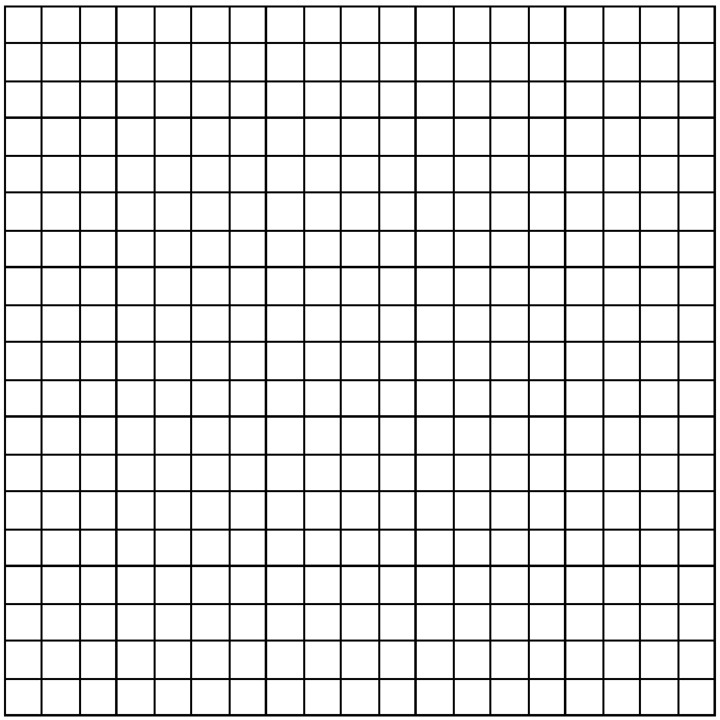
A. Feedback $\lambda_{new} = \begin{cases} \max(0, \lambda_{old} + \varepsilon) & \varepsilon < 0 \\ \min(1, \lambda_{old} + \varepsilon) & \varepsilon > 0 \end{cases}; \qquad \varepsilon = \theta\left(1 - \frac{\|\mathbf{B}_{old}\|_\Omega}{\|\mathbf{B}_{new}\|_\Omega}\right)$

B. l-parameter control $\lambda(\Omega) \rightarrow \arg\min\left(\|\mathbf{B}_{old}\|_\Omega^2 - \|\mathbf{B}_{new}(\lambda(\Omega))\|_\Omega^2\right)^2$

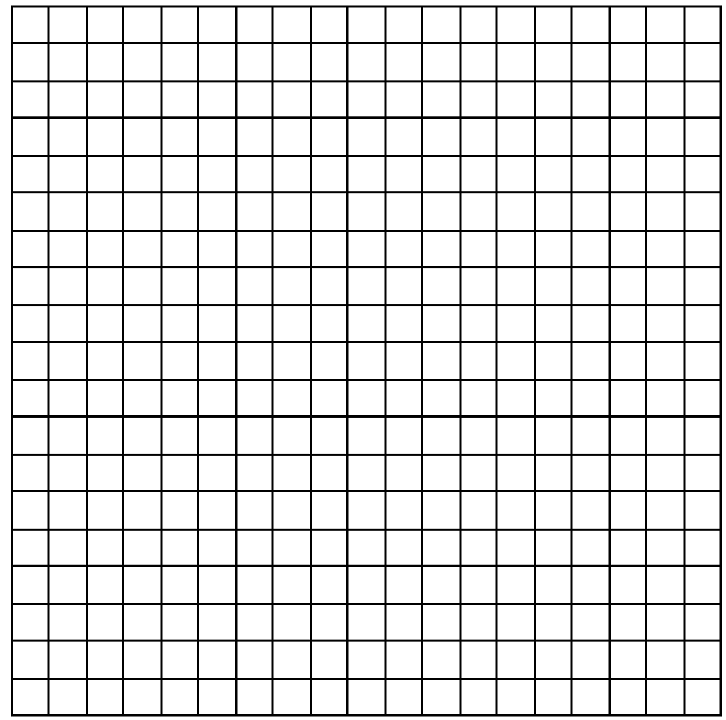
C. Cell-by-cell control $\lambda(K) \rightarrow \arg\min\left(\|\mathbf{B}_{old}\|_K^2 - \|\mathbf{B}_{new}(\lambda(K))\|_K^2\right)^2 \qquad \lambda(\mathbf{P}) = \left(\sum_{\mathbf{P}\in\mathcal{K}} \lambda_{\mathcal{K}}\right) / N(\mathbf{P})$

Numerical Results: Cyclic Remap

Wave Mesh



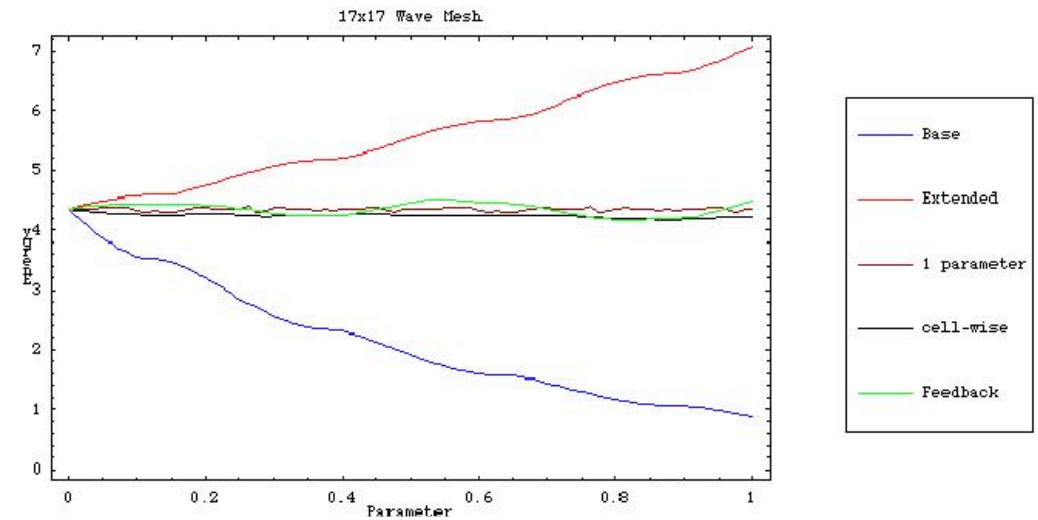
Random Mesh

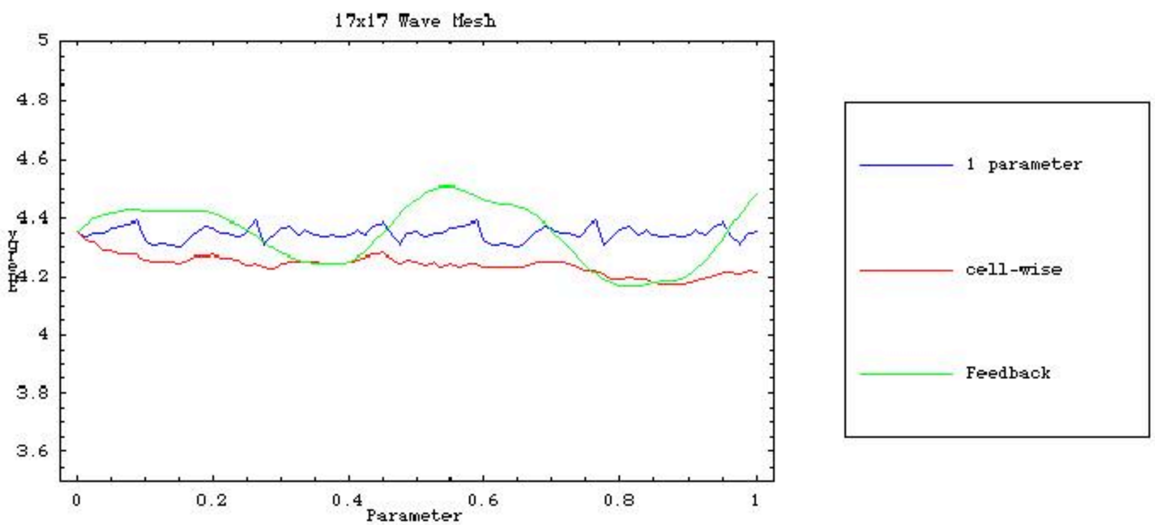


100 cycles

Shashkov and Margolin, LANL LAUR-2002

Potential Optimization: Wave Mesh



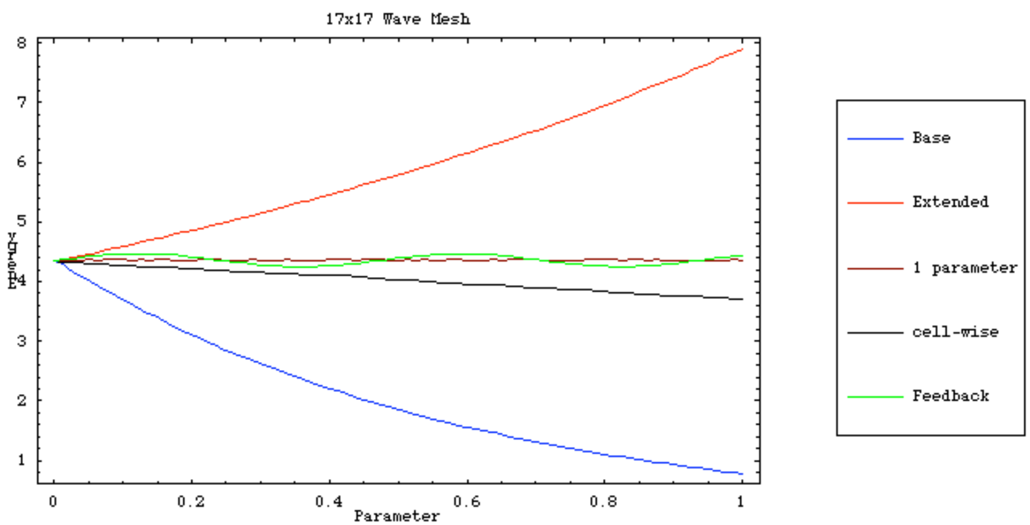


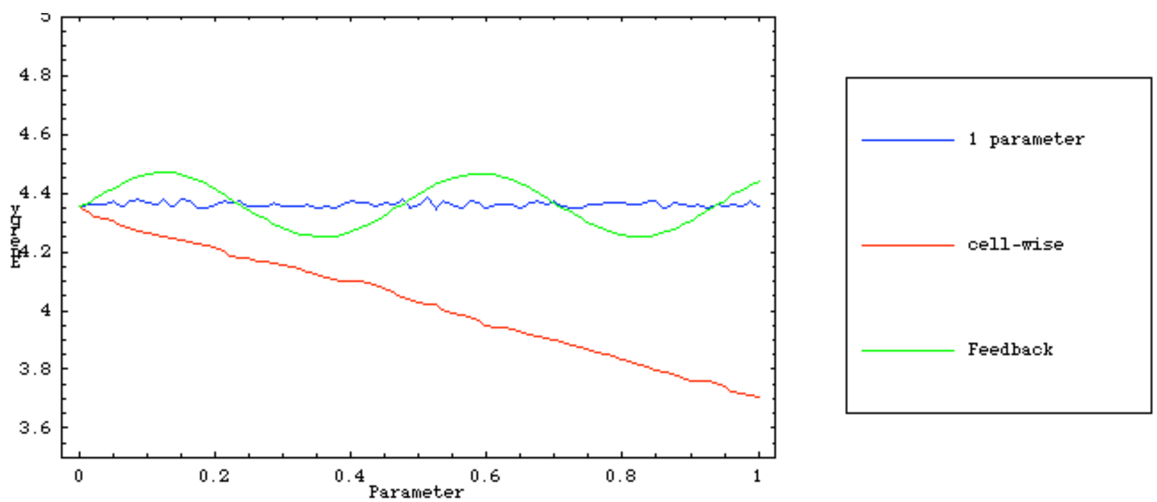
17x17 Wave Mesh: 80 cycles

$$\mathbf{B} = \nabla \times \mathbf{A};$$

$$\mathbf{A} = \sin(2\pi x)\sin(2\pi y)$$

Potential Optimization: Random Mesh





17x17 Random Mesh: 80 cycles

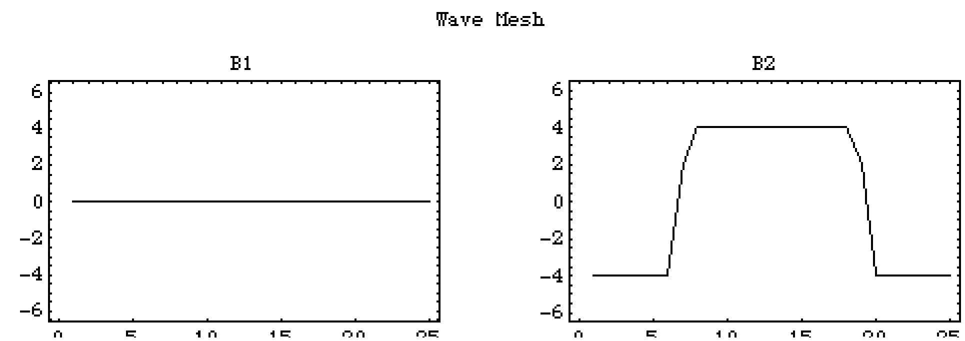
$$\mathbf{B} = \nabla \times \mathbf{A};$$

$$\mathbf{A} = \sin(2\pi x)\sin(2\pi y)$$

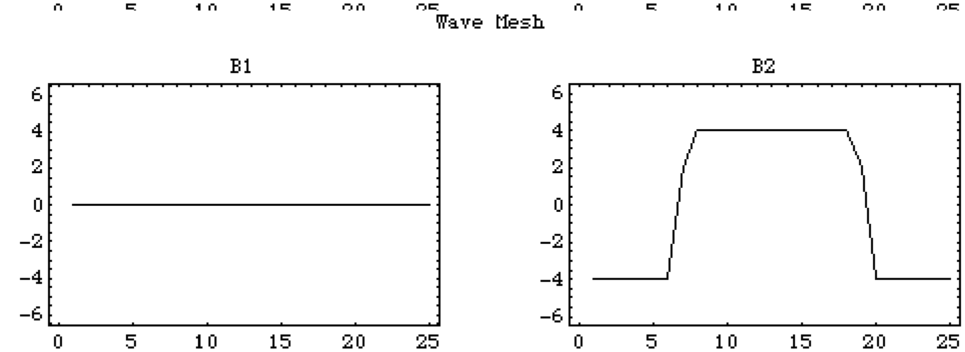
Potential Optimization:
Discontinuous Field

30x30 Wave Mesh:
100 cycles

1-parameter control

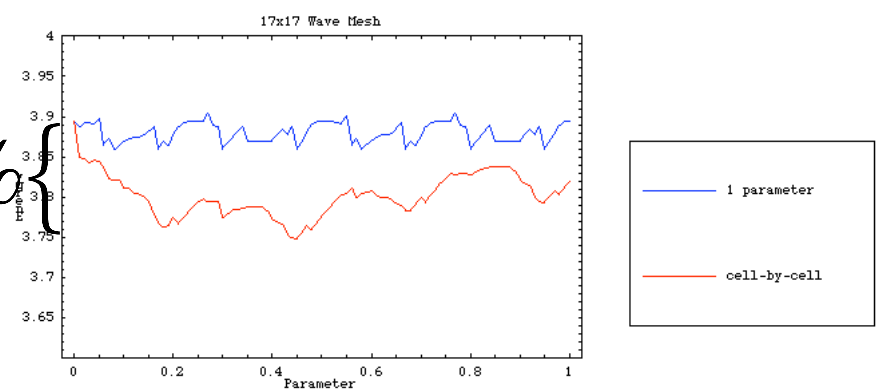


Cell-by-cell control



Energy

< 4%



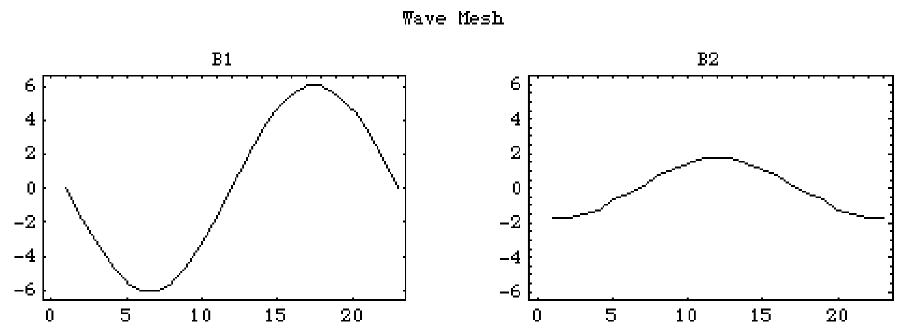
Sandia National Laboratories

CI Base vs. CT Donor:
Smooth Field

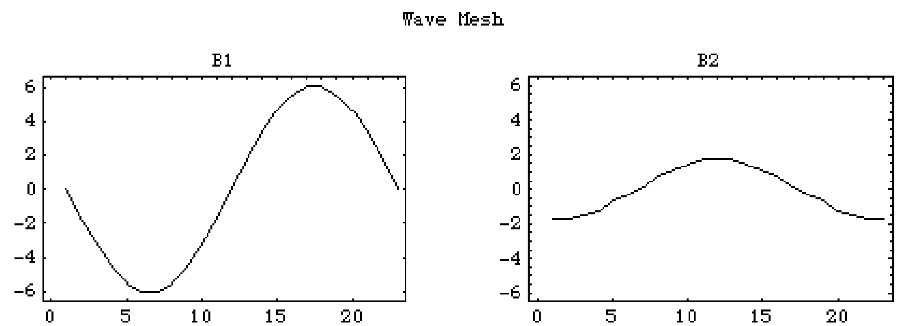
30x30 Random Mesh:
100 cycles

$$\mathbf{B} = \nabla \times \mathbf{A}; \quad \mathbf{A} = \sin(2\pi x)\sin(2\pi y)$$

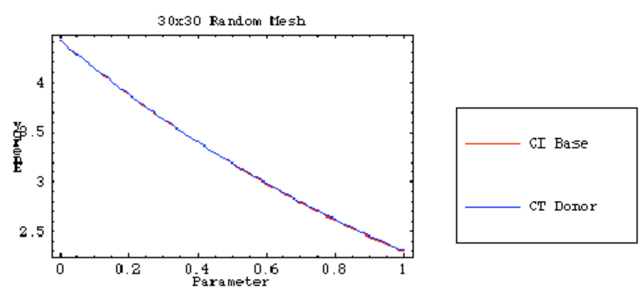
CI Base:



CT Donor:



Energy:

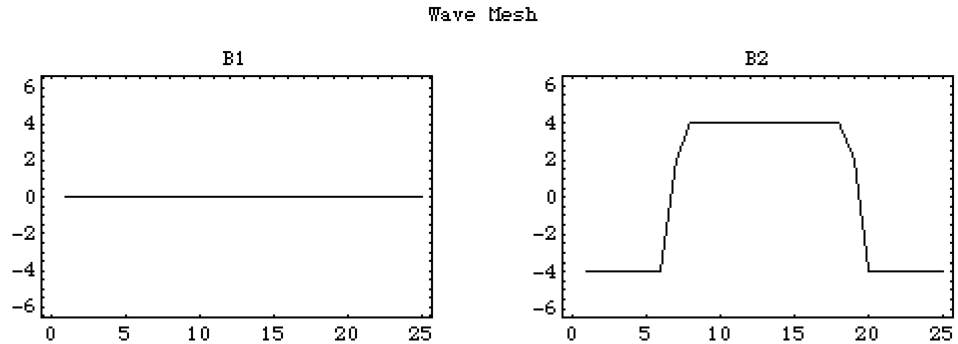


Sandia National Laboratories

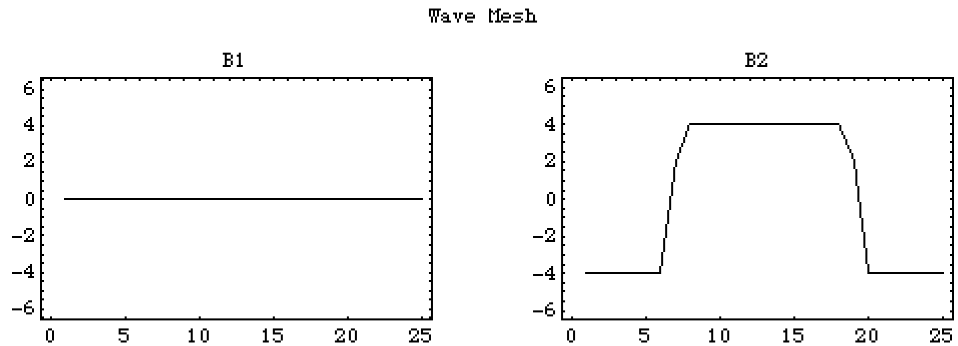
CI Extended vs. CT High Order:
Discontinuous Field

30x30 Random Mesh:
100 cycles

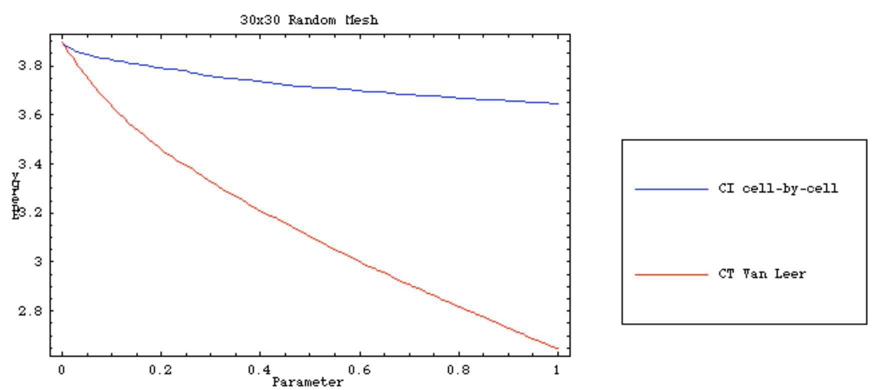
CT Van Leer



CI Cell-by-Cell



Energy:



Sandia National Laboratories

Conclusions

- CI Remap algorithm applicable to arbitrary pairs of grids:
  1. Old and new grid can have different topologies (refinement)
  2. New grid is not necessarily a small perturbation of the old one (no CFL condition required for stability)
  3. High accuracy retained for unstructured grids
  4. Explicit
  5. Local if new grid is small perturbation of the old grid
- Modular: can be extended to different discretizations (FE, FV, FD)

Future work:

- Dissipation control/magnetic energy conservation
- Post-processing of the potential
- 3D algorithm

Constrained Interpolation (CI) Remap